REMARKS

Independent claims 75 and 98 have been amended to further clarify the concept of a partially unspecified term and its corresponding matching restriction.

Support for the language added to the claims can be found in ¶ [0103] of the published specification.

The Examiner has rejected claims 75-99 under 35 USC § 103(a) as being unpatentable over U.S. Pat. No. 6,922,810 (hereinafter "Trower") in view of U.S. Pat. Pub. 2002/0002547 (hereinafter "Sako").

A. Overview of The Trower Reference Teachings

The Trower reference generally teaches a grammar based auto complete user interface system for completing input requests to a variety of computer applications and/or software modules. Trower analyzes incomplete user input and provides one or more auto-complete options that the user can select from to complete the input request (***Abstract – Trower***). Trower's goal is to convert a partially complete input request into a fully complete input request without the user having to specify every character. This is in contrast to the claims of the present invention which describe analyzing a partially unspecified (*but fully complete*) query request so that one or more matches to the partially unspecified input query can be presented to the user as search results (***Abstract – Present Application***). Trower is directed toward a user interface tool for carrying out commands or instructions while the present application is directed toward a query matching search/retrieval tool. Trower seeks to match partial user input with fully specified user requests based on a set of grammar rules. Thus, a partial request in Trower is analyzed and converted to a specific request based on a set of grammar rules. The present application describes and claims a method that analyzes a partially unspecified query having a specific matching restriction associated with the partially unspecified portion of the query, searches one or more documents for matches to the query, and returns the query matching portions of the documents as results to the original query.

A first fundamental difference between Trower and the present application is that Trower analyzes partially complete (or partially incomplete) input requests. The input requests are not queries designed to provide information but are user input commands intended as

input to computer applications or software modules ***(See, c.1, lns. 54-59 - Trower)***. The claims of the present invention, however, describe analyzing fully complete queries that include a partially unspecified *term* with the intent of processing the query against a document repository to return query matches as a result. The query matches provide answers to the original query. A partially unspecified term according to the present application includes a restriction that defines a particular set of character sequences that can match the term ***(See, ¶ [0093] - Present Application)***. The set of terms represented by a partially unspecified term can be defined by characteristics a member must possess to satisfy the associated restriction. Examples of restrictions include categories such as proper name, location, country, date, unit of measurement, company name, baseball players, etc. – ***(See, ¶ [0103] - Present Application)***.

For instance, consider the query "Agatha Christie was born in_[num]" in which _[num] is the partially unspecified term. In this case, a match to the query would require a document to contain a string that included the phrase "Agatha Christie was born in" followed by a number (it could be any number but must be a number) ***(See, ¶ [0094] – [0098] - Present Application)***. However, the context and structure of the query make it such that the likely result will be "Agatha Christie was born in 1890". Compare this to the query "Agatha Christie born in_[location]". This time a match could look like "Agatha Christie was born in England" since "England" fits the restriction defined by the partially unspecified term [location].

A second fundamental difference between Trower and the present application is that Trower's "variables" automatically correspond user input with a category or function while the matching restriction of a partially unspecified term must be verified according to the claims of the present invention. In other words, Trower forces the variable definition onto the user input. In Trower, the rule "open %document%+" where %document% is the wildcard variable would analyze the partial input data "open my resume" and force "my resume" to match the variable whether or not it actually does ***(See, c.10, lns. 7-14 - Trower)***. Whereas, potential matches to the partially unspecified term of a query must satisfy the actual matching restricition as taught and claimed in the present application. If the matching restriction is [num], the document text must contain an actual number ***(See, e.g., ¶ [0094] - Present Application)***. Moreover, Trower's rules are applied to forming complete input requests (not queries) while the claims at issue are directed toward *answering* queries.

Perhaps the best way to distinguish Trower and the present application is to present each system with the same information need. In this case, the user would like to know the current price of Microsoft stock.

In Trower, the user input of "What is Microsoft trading at" would match a pre-defined grammar rule of "What is %symbol%+ trading at" wherein %symbol%+ is nothing more than an unconstrained type of wildcard character. Any character string between the %%+ symbols (including spaces) is accepted without being subjected to a any matching restriction let alone a matching restriction based on syntactical or morphological criteria.

Any purported matching restriction such as "Arithmetic Expression" or "Ticker Symbol" as shown in Fig. 4 do not actually restrict the command/query input. Rather, they are functional indicators to another application that will act on the results of the command/query input. If the user selects "Ticker Symbol" for the example above according to Fig. 4, the term "Microsoft" would be considered a ticker symbol by the currently active application whether it actually is or not. In this case "Microsoft" is not a ticker symbol but merely a company name (MSFT is the ticker symbol for Microsoft). Even then, it is up to the user to make the association between Microsoft or MSFT and ticker symbol. It is not an automated association.

Using the present invention, if the user were searching for the trading price of Microsoft he would enter the query: Microsoft [stock_price]. "Microsoft" is the fully specified portion on the query meaning that all results must include the term Microsoft. [stock_price] is the unspecified portion of the query. The present invention will only return results in which the term Microsoft and an actual stock price are contained in the result.

Thus, the query could return a variety of documents containing search results including:

- "Microsoft closed today at $17.50"
- "Microsoft gained 23 cents to close at 17 ½ "
- "Microsoft (MSFT:$17.50) announced today …"

Referring back to Trower, another partial query (not a partially specified query) such as "What is M" may produce a drop down of selections (results) to the user that include:

- "What is my home address"
- "What is Mr. Smith's telephone number"
- "What is %*TickerSymbol*%+ trading at"

If the user selected the "What is *%TickerSymbol%*+ trading at" option and replaced *%TickerSymbol%*+ with Microsoft, then an application would process the now fully specified command/query. If the application happened to be a typical search engine, the search engine may return results from documents that include the words of the query anywhere in the document as if they were in an OR function (what or is or Microsoft or trading or at) unless the command query were in quotations in which case search results must contain the exact phrase "What is Microsoft trading at". In either case there is no guarantee that a result will be returned that actually contains the data sought, the price of the stock.

B. Overview of The Sako Reference Teachings

The Sako reference generally teaches a computerized method for obtaining additional information on a user selected term (designated character sequence) based on a contextual analysis of the sentence in which the term appears. Sako allows a user to select a term from an electronic document that is being viewed. The term is compared to a knowledge base (via a database retrieval unit) to find a match. For instance, the knowledge base could be an English language dictionary (See, ¶ [0063]). In addition, Sako also analyzes the sentence in which the selected term appears. Sako creates a table of the words in the sentence and includes a column containing the morphological equivalent of each word (e.g., noun, pronoun, verb, etc.). (See, ¶ [0061] and Fig. 3). Keywords are then extracted from the sentence based on their morphological relationship to the designated character sequence and the number of times they appear in the match from the knowledge base for the designated character sequence. (See, ¶ [0079]) For instance, the word "director" is the designated character sequence in the example cited by Sako and the knowledge base is an English language dictionary. The results of the search yield three (3) definitions for 'director' (See, ¶ [0063]) to which the keywords are applied. In this example, definition number two contains the word 'movie' which was identified as a keyword by the morphological analysis performed on the sentence "He is a director of this movie".

The final results, illustrated in Fig. 10 show a callout in the document being viewed giving the definition of the designated character sequence "director".

Thus, Sako provides the user the ability to get more information on a term in a document without having to go elsewhere to search. The term itself is applied to a knowledge base and sentence analysis is used to refine the results on a contextual basis. The morphological

analysis referenced in Sako is used to help select keywords to focus on during the sentence analysis process by keying on the morphological characterization of the words in the sentence that are associated with the designated character sequence.

C. Neither the Trower or Sako References Teach Receiving A Query Containing A Partially Unspecified Term Associated With A Matching Restriction And Determining Matches For The Query Based On The Partially Unspecified Term Associated With A Matching Restriction

With respect to claim 75, the Examiner cites col. 10, lns. 20-26 of Trower as teaching "*receiving a query pertaining to the information need, the query comprised of a fully specified term and a partially unspecified term wherein the partially unspecified term is representative of a matching restriction designed to meaningfully restrict the query results;*"

The above referenced section (col. 10, lns. 20-26 of Trower) has been mis-characterized by the Examiner with respect to the claims of the present application. First and foremost, the user in Trower did *not* enter what the Examiner erroneously considers a partially unspecified term (%symbol%). The *actual* user input in the cited passage is "*what is Microsoft trading at*". This user input does not include any partially unspecified terms. An associated *grammar rule* is comprised of "what is %symbol%+ trading at". The grammar rule is completely separate from the user query and specifies what to do with the user input.

Thus, Trower does not *receive a query* including a partially unspecified term. The grammar rule will equate the term in the user query to the corresponding %symbol%+ term in the grammar rule regardless of the term's actual content. The user could just have easily entered "What is gh%$#jkf trading at" and the grammar rule would automatically equate the nonsensical term gh%$#jkf to the %symbol%+ term even though it is jibberish and attempt to look up a stock price for a non-existent ticker symbol.

There is no disclosure in Trower of a partially unspecified term and, as a result, no disclosure of a matching restriction for the partially unspecified term in a user query.

The second clause of claim 75 requires, "*processing the query against a plurality of information containing documents to determine a query match based on the presence of the fully specified term and the presence of a match to the matching restriction of the partially unspecified term within a document, wherein the order of the fully specified term with respect to the match to the matching restriction of the partially unspecified term need not be in the same order as the query;*". Determining

query matches for the query as described and claimed in the present application involves returning query matches or search results based on a *fully complete but partially unspecified* query. Trower does not determine matches for a query but rather auto-completes user input to create a fully specified user instruction.

Trower has cited col. 10, lns. 1-7 as teaching the above quoted clause. However, that citation merely reinforces that variables are the same as wildcards in that they can be matched by **any** user input meaning that no matching restriction is taught. Wildcards, as explained in the present application, are not the same as a partially unspecified term (*See, ¶ [0108] - Present Application*).

In addition, the cited portion in Trower is referring to the grammar rules. "Actual data that is input by the user that 'matches' the variable is communicated to the corresponding software module for processing." Trower - Col. 10, lns. 5-7.

The cited portions of Trower clearly demonstrate that the user does not provide a partially unspecified term in the actual input query. Nor does Trower determine one or more matches for the query and return such matches from documents searched as results.

Trower's list of options for auto-completing a user input request are not the same as nor equivalent to determining query matches for a query as claimed in the present application.

Sako is cited by the Examiner as broadly teaching the use of morphological analysis in query retrieval systems. However, Sako's use of morphological analysis is limited to equating keywords with parts of speech (See, ¶ [0079]) and it is not used for the purpose of associating a matching restriction to an unspecified portion of a search query.

Thus, the claim language in claim 75 (and 98) describing the present application's use of a morphological restriction has not been taught in Sako.

Moreover, the combination of Trower and Sako would not yield the process that is claimed in the present application. Trower is an input auto-complete process. Sako is an in-line query process that provides additional information on a selected term in a document without having to use a separate application as a search tool. The present application claims a process that returns an answer to a question based on a partially unspecified input query in which the partially input portion of the query is representative of a matching restriction.

The Examiner would modify Trower to include the morphological analysis aspect of Sako. Thus, the Examiner would add a morphological component to Trower. The Examiner

does not, however, describe how such an addition would yield the results claimed in the present application since the goal/result in Trower is not the same as in the present application. Even if such a combination were accepted, it would still not teach the steps and elements of the present claims since neither reference teaches the use of a partially unspecified component having a matching restriction with respect to a user query.

D. The Trower Reference Fails To Teach The Concept Of Indexes, Contexts Within Indexes, And The Use Of Finite State Automatons With Respect To Document Searching

With respect to dependent claims, the Examiner's characterizations of Trower's teachings regarding the claim terms of *indexes*, *contexts*, and *finite state automatons/transducers* are inaccurate and unsupportable using the cited portions and/or the reference as a whole.

As explained and claimed in the present application, an index identifies documents that contain terms. The index may also store the location of documents within a database and information identifying terms that satisfy restrictions (***See, ¶ [0026] - Present Application***). Contexts correspond to linguistically analyzed text strings containing a given term in which the text strings occur within the documents (***See, ¶ [0033] - Present Application***. A finite state automaton (FSA) is a structure having a finite number of states including an initial and final state connected by arcs. The FSA is driven from state to state by a sequence of inputs (***See, e.g., ¶ [0026] and Fig. 24 - Present Application***).

The index, contexts, and FSAs are implemented as database searching techniques that optimize the process of finding query matches to user queries. The Examiner has cited Trower - Col. 10, lns. 20-26 as teaching indexing and contexts within the indexes.

Trower - Col. 10, lns. 20-26 states, *"For example, if a rule in grammar module 230 were "what is %symbol%+ traing at", and if the user entered the data input "what is Microsoft trading", then analysis module 212 would indicate that the rule matches the partial data input, with %symbol%+ matching "Microsoft". If the matching rule were accepted by the user, then the rule "what is Microsoft trading at" would be included in history 232."* This passage merely describes equating or matching the portion of the user input "Microsoft" with the portion of the grammar rule %symbol%+. There is no teaching or disclosure whatsoever of an index identifying documents or contexts within such an index. Trower does not describe interactions with documents in an attempt to search for matches to a user query. Trower describes a software application that operates on

partial user input in order to complete the input request and forward the input request/command to the appropriate software module for execution. Trower is not a tool/application for providing results to search queries and therefore does not interact with a plurality of documents and has no need to create indexes, contexts, or finite state automatons pertaining to the documents and query terms. The terms "index" and "finite" do not appear anywhere in the Trower reference.

In stating that Trower teaches finite state automatons and finite state transducers, the Examiner merely cites to and equates the term "computer" as set forth in col. 2, lns. 49-52 to a finite state automaton (FSA). Such a characterization is woefully inadequate and inaccurate given the present application's description of the terms finite state automaton (FSA) and finite state transducer (FST) in paragraphs [0172] and [0184] respectively.

III. Conclusion

In sum, neither Trower nor Sako receive a query containing a partially unspecified term associated with a matching restriction. Moreover, neither Trower nor Sako determine matches for the query based on the partially unspecified term associated with a matching restriction. Nor does Trower teach the concept of indexes, contexts within indexes, and the use of finite state automatons with respect to document searching. Rather, Trower endeavors to determine the remainder of a user input request based on (i) the partial input, (ii) a set of grammar rules, (iii) the user's recent data entry history, (iv) a user context, and (v) a scoring process. (Trower – col. 3, lns. 29-35). And, Sako's use of morphological analysis is limited to equating keywords with parts of speech (See, ¶ [0079]). It is not used for the purpose of associating a matching restriction to an unspecified portion of a search query.

Applicant has shown that there are clear errors in the Examiner's rejections as well as omissions of one or more essential elements needed for a prima facie rejection. As such, applicant respectfully requests reconsideration and withdrawal of the 35 USC §103(a) rejection based on Trower in view of Sako. It is applicants' belief that the references cited in combination do not teach all the elements and limitations recited in the claims as required under 35 USC §103(a).

Applicant believes that all of the Examiner's objections and rejections have been addressed and overcome and requests that all such objections and rejections be withdrawn.

Respectfully submitted,

Date: Aug. 23, 2010

//GREGORY STEPHENS//

Gregory Stephens
Attorney for Applicants
Registration No. 41,329
Williams Mullen PC
PO Box 12109
Research Triangle Park, NC 27709-3646
(Tel) 919.981.4318
(Fax) 919.361.2262
gstephens@williamsmullen.com